

Tecnólogo em Análise e Desenvolvimento de Sistemas

RECUPERAÇÃO BIMESTRAL – 1.º Bimestre

PROGRAMAÇÃO DE COMPUTADORES II

Paranaguá, 22 de Dezembro de 2016

Gil Eduardo de Andrade

Avaliação Prática

Tempo Total: 150 minutos

INSTRUÇÕES DA RESOLUÇÃO DA AVALIAÇÃO

1. A avaliação é individual e com consulta a livros, materiais didáticos e códigos-fonte. Não é permitido o uso de variáveis globais para resolução das questões.
2. Não existem questões parcialmente corretas, ou seja, como cada questão equivale a 25% do conceito final, ou o aluno atinge os 25% na questão ou ela é desconsiderada.
3. O professor não tirará dúvidas com relação a possíveis erros de compilação ou sintaxe, por isso todas as questões possuem exemplos de entrada com sua correspondente saída.
4. Ao término do tempo de prova o professor recolherá, via *pen drive*, os códigos-fonte em C de resolução das questões, estes devem estar nomeados de acordo com o exercício que resolvem, ou seja, ex01.c, ex02.c, ex03.c e ex04.c.

PONTEIROS / FUNÇÕES / PASSAGEM DE PARÂMETRO

- 1) (**Médio**) Faça uma função em C que receba dois valores inteiros (*int*) e retorne um vetor contendo todos os números primos existentes entre os dois números passados como parâmetro. A posição zero do vetor retornado deve conter o total de números primos encontrados. Dentro da função **main** exiba os primos contidos nesse vetor. Veja o exemplo:

```
g1l3du4rd0@asus-ultrabook-g1l:~/Documentos/Rec_3B
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$ ./a.out
- Valor inicial: 3
- Valor final: 15
- Primos entre 3 e 15: 3 5 7 11 13
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$
```

- 2) (**Fácil**) Faça uma função em C que receba um vetor inteiro aleatório (gerado e apresentado na função **main**) com valores entre 10 e 99 e um inteiro contendo o seu tamanho total. O tamanho do vetor deve ser definido pelo usuário entre 10 e 20 elementos. A função deve encontrar todos os números múltiplos de 2 e 3 e retornar a soma desses valores. Veja o exemplo:

```
g1l3du4rd0@asus-ultrabook-g1l:~/Documentos/Rec_3B
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$ ./a.out
- Tamanho do vetor (entre 10 e 20): 10
- Vetor aleatório: 72 32 25 18 93 78 53 34 80 20
- Soma dos múltiplos de 2 e 3: 168
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$
```

- 3) (**Fácil**) Faça uma função em C, sem retorno, que receba um vetor (20 elementos) de caracteres aleatório, que não se repetem, contendo apenas letras maiúsculas de A-Z. Essa função deve contabilizar o número total de vogais e substituir todas as consoantes pelo caractere **hashtag**. O valor referente ao total de vogais encontradas deve ser enviado para função **main** via passagem de parâmetro por referência, ou seja, além do vetor aleatório a função deve receber também um segundo parâmetro. A função **main** deve exibir o vetor atualizado (com as **hashtags**) e o valor total de vogais que foi armazenado na variável passada por referência. Veja o exemplo:

```
g1l3du4rd0@asus-ultrabook-g1l:~/Documentos/Rec_3B
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$ ./a.out
- Vetor aleatório: C N D O S B M H T J P Q E Z X I V A U K
- Vetor modificado: # # # O # # # # # # # # E # # I # A U #
- Total de vogais: 5
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$
```

- 4) (**Difícil**) Utilizando as bibliotecas “*pilha.h*” e “*fila.h*” (implementadas na atividade da aula do dia 22/setembro) codifique uma função que receba um ponteiro char com tamanho definido pelo usuário (entre 5 e 10 caracteres). Esse ponteiro deve ser preenchido com caracteres aleatórios minúsculos e maiúsculos entre A-Z e a-z e apresentado pela função **main**. Ao receber o vetor e o seu tamanho a função deve empilhar as consoantes minúsculas e vogais maiúsculas (na pilha) e enfileirar as consoantes maiúsculas e as vogais minúsculas (na fila). Após isso, a função deve desempilhar um caractere da pilha e desenfileirar um caractere da fila repetidamente, até que a pilha e a fila fiquem vazias. Todos os caracteres removidos dessas estruturas devem ser armazenados num vetor (dentro da função) na ordem em que foram retirados. Por fim, o vetor em questão deve ser retornado para função **main** e apresentado. Veja o exemplo:

```
g1l3du4rd0@asus-ultrabook-g1l:~/Documentos/Rec_3B x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$ ./a.out
- Tamanho do vetor (entre 5 e 10): 5
- Vetor aleatorio: l f M b V
- Vetor modificado: b M f V l
[g1l3du4rd0@asus-ultrabook-g1l Rec_3B]$
```



TABELA DE CÓDIGOS ASCII – MATERIAL DE APOIO

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	##32;	Space	64	40	100	##64;	@	96	60	140	##96;	^
1	1	001	SOH (start of heading)	33	21	041	##33;	!	65	41	101	##65;	A	97	61	141	##97;	a
2	2	002	STX (start of text)	34	22	042	##34;	"	66	42	102	##66;	B	98	62	142	##98;	b
3	3	003	ETX (end of text)	35	23	043	##35;	#	67	43	103	##67;	C	99	63	143	##99;	c
4	4	004	EOT (end of transmission)	36	24	044	##36;	\$	68	44	104	##68;	D	100	64	144	##100;	d
5	5	005	ENQ (enquiry)	37	25	045	##37;	%	69	45	105	##69;	E	101	65	145	##101;	e
6	6	006	ACK (acknowledge)	38	26	046	##38;	&	70	46	106	##70;	F	102	66	146	##102;	f
7	7	007	BEL (bell)	39	27	047	##39;	'	71	47	107	##71;	G	103	67	147	##103;	g
8	8	010	BS (backspace)	40	28	050	##40;	(72	48	110	##72;	H	104	68	150	##104;	h
9	9	011	TAB (horizontal tab)	41	29	051	##41;)	73	49	111	##73;	I	105	69	151	##105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	##42;	*	74	4A	112	##74;	J	106	6A	152	##106;	j
11	B	013	VT (vertical tab)	43	2B	053	##43;	+	75	4B	113	##75;	K	107	6B	153	##107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	##44;	,	76	4C	114	##76;	L	108	6C	154	##108;	l
13	D	015	CR (carriage return)	45	2D	055	##45;	-	77	4D	115	##77;	M	109	6D	155	##109;	m
14	E	016	SO (shift out)	46	2E	056	##46;	.	78	4E	116	##78;	N	110	6E	156	##110;	n
15	F	017	SI (shift in)	47	2F	057	##47;	/	79	4F	117	##79;	O	111	6F	157	##111;	o
16	10	020	DLE (data link escape)	48	30	060	##48;	0	80	50	120	##80;	P	112	70	160	##112;	p
17	11	021	DC1 (device control 1)	49	31	061	##49;	1	81	51	121	##81;	Q	113	71	161	##113;	q
18	12	022	DC2 (device control 2)	50	32	062	##50;	2	82	52	122	##82;	R	114	72	162	##114;	r
19	13	023	DC3 (device control 3)	51	33	063	##51;	3	83	53	123	##83;	S	115	73	163	##115;	s
20	14	024	DC4 (device control 4)	52	34	064	##52;	4	84	54	124	##84;	T	116	74	164	##116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	##53;	5	85	55	125	##85;	U	117	75	165	##117;	u
22	16	026	SYN (synchronous idle)	54	36	066	##54;	6	86	56	126	##86;	V	118	76	166	##118;	v
23	17	027	ETB (end of trans. block)	55	37	067	##55;	7	87	57	127	##87;	W	119	77	167	##119;	w
24	18	030	CAN (cancel)	56	38	070	##56;	8	88	58	130	##88;	X	120	78	170	##120;	x
25	19	031	EM (end of medium)	57	39	071	##57;	9	89	59	131	##89;	Y	121	79	171	##121;	y
26	1A	032	SUB (substitute)	58	3A	072	##58;	:	90	5A	132	##90;	Z	122	7A	172	##122;	z
27	1B	033	ESC (escape)	59	3B	073	##59;	;	91	5B	133	##91;	[123	7B	173	##123;	{
28	1C	034	FS (file separator)	60	3C	074	##60;	<	92	5C	134	##92;	\	124	7C	174	##124;	
29	1D	035	GS (group separator)	61	3D	075	##61;	=	93	5D	135	##93;]	125	7D	175	##125;	}
30	1E	036	RS (record separator)	62	3E	076	##62;	>	94	5E	136	##94;	^	126	7E	176	##126;	~
31	1F	037	US (unit separator)	63	3F	077	##63;	?	95	5F	137	##95;	_	127	7F	177	##127;	DEL

Source: www.LookupTables.com